02

# Agregando dependencias

## Transcripción

[00:00] Bienvenidos a la clase número 3 de su curso de Spring Boot Rest API. Hasta el momento ya hemos visto cómo enviar datos desde el cliente, en este caso puede ser una aplicación front end, mobile o incluso nuestro cliente de Insomnia, a nuestro API, a nuestro back end.

[00:18] Ya estamos recibiendo los datos aquí y lo que estamos haciendo por ahora es solamente imprimirlo. Es un buen inicio, pero aún queda mucho más por hacer. Lo que vamos a hacer en esta tercera clase son dos cosas básicamente. Lo primero si tú recibes los datos aquí tienes que hacer algo con ellos.

[00:34] Entonces lo que pueda ser sería guardado en la base de datos, enviarlo a otro API, etcétera. Nosotros vamos a centrarnos en guardarle en una base de datos. Para eso vamos a ver nuestras dependencias que tenemos hasta este momento.

[00:49] Vemos que aquí hasta ahora tenemos el spring-web que hemos estado usando, el DevTools para que no reinicie el servidor automáticamente. Lombok, que no es una librería de Spring propiamente dicho pero Spring Integra muy bien dado que es ampliamente usado en la mayoría de aplicaciones y tenemos el Spring-starter-test.

[01:17] Esto ya viene por defecto porque Spring, por buenas prácticas, tenemos que testear el código, no lo estamos haciendo ahora porque nos estamos entrando 100% e implementar el API y por fines didácticos estamos saltando

esta parte. Hay un curso en Alura sobre pruebas unitarias, usando de JUnit, altamente recomendado. Entonces si quieren profundizar sus conocimientos en testing, pues es uno de los mejores cursos que pueden ver. Vamos a volver al tema.

[01:47] Ya vimos que tenemos por ahora solamente cuatro dependencias, pero si queremos trabajar con bases de datos, necesitamos algo más. Existen formas de agregar dependencias en Spring. La primera, yo podría copiar esto, pegarlo y escribir el nombre del groupId y el artifactId, el artefacto la librería que yo quiero usar, pero es muy expuesto a errores humanos, yo me puedo equivocar escribiendo, Spring puede que no encuentre la librería porque está mal escrita, la versión es equivocada, entonces nos exponemos a mucho.

[02:24] Existe una forma de agregar dependencias en Spring Boot muy interesante y es la que vamos a ver ahora. Recuerda nuestro amigo Spring Initializr. Aquí podemos generar el proyecto pero también podemos hacer un pequeño truco para agregar dependencias. Miren.

[02:42] Si yo le doy aquí en add dependencies por ejemplo yo sé que una de las dependencias que voy a utilizar es Spring data. Entonces voy a agregar Spring data JPA. Vemos que lo agrega aquí. Perfecto. Como base de datos yo voy a usar MySQL, entonces voy a decirle: “Por favor instala el MySQL driver”. Se me pasó una T. Listo, MySQL driver. Tenemos MySQL driver.

[03:09] Y vamos a instalar también Flyway. Flyway es un gestor de migraciones de base de datos. Vamos a profundizarlo en los próximos videos, no se preocupen, pero es básicamente para mantener tu base de datos como código versionado propiamente dicho. No se preocupen si no lo entienden por ahora, lo van a entender con certeza más adelante.

[03:29] Entonces agregamos Flyway Migration. Y por el momento yo creo que ya estamos okay con estas tres dependencias, no vamos a usar nada más. Nuevamente, Flyway Migration no es parte de Spring Boot, no es propia de

Spring pero al igual que MySQL Driver Spring las integra muy bien porque son ampliamente usadas en el mercado y la idea de Spring es proporcionar dos herramientas lo más fácil posible.

[03:56] En este caso yo no le voy a dar clic a generate. Yo le voy a dar clic a explore. ¿Por qué? Porque si le das clic aquí lo que esto te va a dar es un overview de cómo se va a generar el proyecto. Lo que estamos viendo aquí, por ejemplo es que me está botando un archivo de build.gradle.

[04:18] Esto porque yo no seleccioné el gestor de dependencias de Maven, entonces lo que yo voy a hacer aquí es cerrar porque no es el formato en el que yo deseo descargar mis dependencias, le voy a dar close y aquí voy a seleccionar Maven. Si le doy explore, como pueden ver es prácticamente el proyecto que se me va a descargar, pero esta es una vista previa de lo que voy a tener.

[04:39] Y lo que voy a hacer aquí, simplemente es copiar estas cuatro dependencias de aquí, porque la de test ya la tenemos y no hay ninguna otra más. Yo no he agregado ninguna otro más entonces lo que hago es copiar, regreso aplicación y aquí abajo voy a pegar esas dependencias y voy a guardar.

[05:06] ¿Cómo estoy guardando? Normalmente IntelliJ puede que si está configurado dispare automáticamente un proceso de sincronización de dependencias porque como pueden ver está en rojo porque no las encuentra, no están descargadas aquí localmente, pero si no también podemos darle al botón de Maven y un refresh.

[05:26] Con el refresh vemos que aquí está diciendo que está resolviendo las dependencias del API. Esto quiere decir está descargando las dependencias. Hasta ahora tenemos solamente cuatro. Mientras descarga entonces vamos a hacer un corte en este video y vamos a retornar cuando ya estén todas las dependencias aquí.

[05:46] Bien. Todas las dependencias ya están descargadas y como podemos ver aquí, ya tenemos el árbol completo. Tenemos ya las dependencias de Flyway, tenemos el driver de MySQL, tenemos Spring data, entonces ya estamos listos para comenzar a trabajar.

[06:04] Yo ya tengo un servidor de MySQL corriendo aquí en este momento. Como este no es un curso de base de datos, entonces no se preocupen, no vamos a profundizar mucho en lo que es instalación de MySQL. Es el típico MySQL que todos ustedes conocen. En este caso yo he descargado una versión para mi computador, ustedes pueden hacer lo mismo.

[06:24] Pueden usar cualquier programa cliente. Pueden usar DBeaver, MySQL Workbench, el programa que se sientan más cómodos de utilizar. En mi caso, como yo tengo una versión Premium de IntelliJ, yo tengo un plugin aquí que me permite acceder a la base de datos.

[06:40] Entonces yo ya tengo mi base de datos configurada, tengo todo. Solo para ver si mi proyecto compila correctamente, yo le voy a dar Play para iniciar el proyecto. Y vamos a ver qué es lo que pasa, vemos que inicia, Spring está comenzando y vemos que tiene un error. ¿Cuál es el error?

[07:00] Lo que me dice es que la aplicación falló al iniciar porque no hay configurado un data source. ¿Y eso por qué sale? Porque Spring data por defecto, si es que yo ya tengo la dependencia instalada, lo primero que va hacer Spring data es buscar por una conexión con una base datos. ¿Cómo soluciono esto? Con el archivo properties.

[03:22] Voy aquí a resources y voy a application.properties y aquí yo necesito agregar tres properties para que pueda funcionar el springdata. Por ejemplo, la primera es Spring.datasource. IntelliJ me va a ayudar aquí, datasource, no quiero hikari, bueno. Voy a copiar porque creo que es más rápido así, url. ¿Cuál es la url?

[07:52] Mi datasource comienza como siempre con jdbc:mysql://localhost/vollmed-api. Vamos a usar ese nombre. Segundo, necesitamos spring.datasource.username. En mi username yo voy a usar root que es el que tengo por defecto configurado y aquí voy a copiar esto. Acá solamente voy a cambiar por password y mi contraseña que yo le configure a mi mysql local es del 1 al 8.

[08:35] Entonces con estas tres propiedades yo ya tengo listo mi configuración con la base de datos. Vamos a ver si esto es verdad. Entonces vamos a iniciar el servidor nuevamente porque falló al iniciar. Entonces lo vamos a iniciar y vamos a ver qué es lo que pasa.

[08:56] En efecto, vemos que no podemos iniciar y ahora tenemos un error diferente. Esto ya no es un error de inicio. Esto es una exception. Vamos a ver qué es lo que dice: Unknown database 'vollmed-api'. ¿Esto por qué es? Porque en efecto esto es bueno porque ya tengo la conexión con la base de datos, pero no existe la base de datos 'vollmed-api', tengo que crearla.

[09:21] Para eso, entonces lo que yo voy a hacer es nuevamente ustedes pueden usar algún cliente de base de datos, el que ustedes quieran, el que más le guste. Yo voy a usar el que tengo ya integrado en esta versión de IntellyJ. Entonces lo que voy a venir es voy a entrar aquí, y voy a abrir una nueva consola para queries. La abro aquí.

[09:45] Me abre una terminal para ejecutar queries y le voy a decir create database y voy a copiar el nombre de la base datos que yo especifiqué aquí que es voll-médico.api. Entonces copiamos, pegamos y ejecutamos, vamos a ver qué es lo que nos dice. Okay, tengo un error por la sintax-api. ¿Por qué? Porque hay un error aquí.

[10:11] Entonces lo que hago es ponerle un guión bajo porque mi sintaxis está equivocada, entramos, vemos que lo creó. Vamos a ver ahora qué sucede. Y listo. Create voll-med.api, una fila afectada en 10 milisegundos. Perfecto.

Entonces ahora voy a iniciar nuevamente mi servidor y vamos a ver qué es lo que sucede ahora.

[10:42] Iniciando y tenemos otra excepción, porque claro, yo actualicé mi nombre de la base de datos, tengo que actualizarlo también en mi archivo properties. Como pueden ver, tiene que estar altamente sincronizado. Le damos guardar, le damos play de nuevo, porque siempre Spring data JPA lo primero que hace es escanear y verifica que en efecto tu base de datos exista. Y ya tenemos la aplicación inicializada con la base de datos.

[11:15] Si vemos aquí vemos que en efecto ya tiene otros logs agregados en el inicio. Está usando el dialecto de MySQL con Hybernate. Esto lo vamos a explorar más adelante. No se preocupen. Pero al final vemos que aquí está iniciando la conexión con la base de datos. Entonces ya estamos listos para configurar lo que es la capa de persistencia con nuestro API. Nos vemos en el siguiente video.